

MATERIA	Programación			
EVAL.	2	BLOQUE	EVALUACIÓN	
ALUMNO			GRUPO	DAW

EXAMEN	
I.E.S. CASTELAR	
DPTO	IFC

28 de febrero de 2025

Crea un proyecto maven en INTELLIJ (si no lo tienes ya) en `{{REPO}}/prog/examenes` (ahí debe aparecer un `pom.xml`). En la ubicación por defecto (`src/main/java`) deberás tener el paquete `ut8` con la clase `Examen2aEvaluacion` que los métodos de clase siguientes:

RA1, 2, 3, 4 y 6:

- [1,5p] Arrays 1 dimensión -> Implementa el método `void ordenarSelf(int[] vector)`, que reciba un array de 1 dimensión de enteros y lo devuelva ordenado. (NO puedes utilizar la clase `ARRAYS`).

- [3p] Arrays 2 dimensiones -> Implementa el método `int[][] obtenerEnteros(char[][] matriz)`, que reciba una matriz de caracteres y devuelva una de iguales dimensiones solo con los números enteros de la matriz original.

Indica brevemente (2 líneas) como implementarías el método `obtenerNoEnteros` que devolvería exactamente la matriz complementaria a la anterior.

- [2p] ROT13 -> Implementa el método `String rot13(String cod)` que reciba un String y codifica cada uno de sus caracteres con el método Cesar 13.

Este método consiste en desplazar el abecedario (inglés) 13 posiciones exactamente, con lo que la letras abc pasan a ser las nop y así sucesivamente, y donde en el alfabeto inglés coincide además que los caracteres nop pasan a ser los abc nuevamente, por lo que el mismo método podremos utilizarlo para codificar y decodificar el mensaje. El resto de caracteres (espacios, comas, interrogantes, etc) se dejan sin codificar.

Puedes probar el método con el siguiente String: `¿Como llevo el examen? ... chrfr cnerpr dhr inf ovra.`

Recuerda que la clase `Character` dispone del método `static boolean isLetter(char ch)` que devuelve `true` si le pasas una letra y `false` en cualquier otro caso.

Se recomienda implementar el método auxiliar `char rot13(char c)`.

- [3p] Colecciones -> Se desea registrar las faltas de los alumnos de la clase. Realiza una implementación que nos proporcione los métodos de instancia `int registrarFalta(String nombre, String fecha)` que devuelve el número de faltas totales del mismo, y `String mostrarFaltas()` que devolverá el listado de días separados por espacios " " en formato `'DD-MM-AAAA'`.

RA1, 2, 5 y 7:

- [1p] Herencia: Crea una clase `Persona` con los atributos `nombre` y `edad` (entero). También la clase `Alumno` que es de tipo `Persona` y tiene además el atributo `numeroDeMatricula` (entero).

- [1p] Crea la interfaz `DAO` con los métodos:

- `List<Persona> getAll(String rutaCompletaAarchivo)`.
- `int writeAll(String rutaCompletaAarchivo, List<Persona> listaDePersonas)`

- [1p] Haz que `Persona` sea ordenable de forma natural del más joven al más mayor. Crea:

- [2p] **Opción A):** `ArchivoDaoImp` que implementa el DAO mediante serialización de objetos básica.

- [3p] **Opción B):** `ArchivoDaoImpJson` que implementa el DAO mediante serialización de objetos en formato Json.

Nota 1: No es evaluable si no se tienen más del 80% de las actividades entregadas.

Nota 2: Deberéis trabajarlo OBLIGATORIAMENTE como proyecto maven (`pom.xml`) y subir la estructura indicada a vuestro repositorio de GitHub según la ruta indicada. Cualquier alteración de ubicación o nombre implicará la calificación de 0 puntos.

Nota 3: Se evaluará sólo los métodos que estén verificados mediante pruebas unitarias.

Nota 4: Debéis superar (>5) todos los RA para superar el examen.

Nota 5: El desarrollo completo del examen debe estar documentado en vídeo mediante vokoscreen o aplicación similar.